**Automate Your Infrastructure Using Ansible Course Quiz**

**Module 2 – Ansible Introduction**

1. **Which network protocol does Ansible use to communicate between the nodes?**
2. **FTP**
3. **HTTPS**
4. **SSH**
5. **SNMP**

The correct answer is option c - SSH.

Explanation for correct answer:

* Ansible uses the SSH protocol to communicate between nodes. Ansible is designed to be agent-less - which means Ansible does not need a central server or agents on the target nodes. All communications between the control Ansible node and the target network nodes happens SSH.

Explanation for Wrong answer:

* Option a - File Transfer Protocol (FTP), is not a protocol which Ansible uses to communicate with the target network nodes.
* Option b - Hypertext Transfer Protocol (HTTP), is not a protocol which Ansible uses to communicate with the target network nodes.
* Option d - Simple Network Management Protocol (SNMP), is not a protocol which Ansible uses to communicate with the target network nodes.

1. **Is this statement true – “Ansible can be configured to communicate over a custom SSH network port rather than the default port 22”.**

The correct answer is True.

Explanation for correct answer:

* Yes, this statement is true. We can configure Ansible to communicate on a customer SSH port other than 22. This parameter can be configured by changing remote\_port parameter in Ansible configuration file i.e., ansible.cfg. This is useful in scenarios where specific ports are needed for SSH for security reasons.

Explanation for Wrong answer:

* That is not correct.

1. **What is Ansible inventory?**
2. **A list of task names**
3. **A list of module names**
4. **A list of host names**
5. **A list of user names**

The correct answer is option c - A list of host name.

Explanation for correct answer:

* Ansible inventory is a collection of host names which can be grouped together. Ansible uses this inventory file to keep track of the network hosts for which a configuration change needs to be applied.

Explanation for Wrong answer:

* Option a – Ansible tasks are responsible for calling a module with a speciﬁc set of parameters. They allow us to call Ansible modules.
* Option b - Ansible Module are used to control or manage resources on local or remote servers.
* Option d - A list of user names is not directly related to Ansible.

1. **Which Ansible configuration location has the highest priority?**
2. **ANSIBLE\_CONFIG environment variable**
3. **The current folder**
4. **Home folder**
5. **/etc/ansible/ansible/cfg**

The correct answer is option a - ANSIBLE\_CONFIG environment variable.

Explanation for correct answer:

Ansible configuration can be stored in multiple locations. Ansible reads the configuration in a priority order. The order of priority is

* ANSIBLE\_CONFIG (an environment variable)
* ansible.cfg (in the current directory)
* .ansible.cfg (in the home directory)
* /etc/ansible/ansible.cfg

By default, the Ansible\_CONFIG environment variable it takes as a top priority

Explanation for Wrong answer:

* Option b, c and d - By default the ANSIBLE\_CONFIG environment variable takes the top priority

1. **By default Ansible configuration file is located at**
2. **/etc/ansible/hosts**
3. **/ansible/etc /hosts**
4. **/hosts/etc/ansible**
5. **/etc/hosts/ansible**

The correct answer is option a - /etc/ansible/hosts.

Explanation for correct answer:

* At the time of installation, by default Ansible configuration file has being saved in the location /etc/ansible/hosts.

Explanation for Wrong answer:

* Option b, c and d – there is no such location in Ansible. At the time of installation, by default Ansible configuration file has being saved in the location /etc/ansible/hosts.

**Module 3: Run and Execute Ansible Tasks**

1. **\_\_\_\_\_\_\_\_ is an agent-less tool that allows us to perform deployment and configuration of multiple nodes from a single system**
2. **AWS**
3. **Ubuntu**
4. **CentOS**
5. **Ansible**

The correct answer is option d - Ansible.

Explanation for correct answer:

* Ansible is an agent-less tool that allows us to quickly and efficiently manage multiple machines, also known as nodes and allows us to perform deployment and configuration for these nodes from a single system.

Explanation for Wrong answer:

* The option a, b, and c AWS is not an automation nor configuration management tool.
* Ubuntu is not an automation nor configuration management tool.
* CentOS is not an automation nor configuration management tool.

1. **\_\_\_\_\_\_\_\_\_\_ is a set of sequential tasks similar to a shell script or bash script.**
2. **Ansible playbook**
3. **Ansible ad-hoc commands**
4. **Ansible inventory**
5. **Ansible Galaxy**

The correct answer is option a - Playbook.

Explanation for correct answer:

* Ansible playbook is a set of sequential tasks. Instead of running each task one by one manually - we write the commands in a playbook. This is like a shell script or bash script.

Explanation for Wrong answer:

* Option b – Ansible ad-hoc commands allow you to execute simple tasks at the command line against one or all of your hosts.
* Option c - Ansible inventory is a collection of host names. Ansible uses this inventory file to keep track of the target network hosts for which a change needs to be applied.
* Option d - Ansible Galaxy refers to a community website where users can share Ansible roles.

1. **Is this statement true:**

**“If you are not using any Ansible facts in your tasks, you can disable the “fact gathering” step for improved speed”**

The correct answer is True.

Explanation for correct answer:

* Yes, this statement is true. If you are not using any Ansible facts in your tasks, you can disable the “fact gathering” step for improved speed. To do so, simply add the property gather\_facts: False in your playbook. As Ansible uses "Facts", which is mainly a system environment information. It gathers context before running the Tasks. Ansible uses these facts to check state and see if it needs to change anything in order to get the desired outcome. This makes it safe to run Ansible Tasks against a server over and over again but If you are not using any Ansible facts in your tasks, you can disable the “fact gathering” step for improved speed.

Explanation for Wrong answer:

* No, this statement is true. If you are not using any Ansible facts in your tasks, you can disable the “fact gathering” step for improved speed. To do so, simply add the property gather\_facts: False in your playbook. As Ansible uses "Facts", which is mainly a system environment information. It gathers context before running the Tasks. Ansible uses these facts to check state and see if it needs to change anything in order to get the desired outcome. This makes it safe to run Ansible Tasks against a server over and over again but If you are not using any Ansible facts in your tasks, you can disable the “fact gathering” step for improved speed.

1. **What option can we use to pass parameters to an Ansible module when we run Ansible ad-hoc commands?**
2. **-param**
3. **-a**
4. **-p**
5. **–args**

The correct answer is option b - -a.

Explanation for correct answer:

* We can specify module parameters with the “–a” option. multiple parameters can be passed by separating them with space.

Example:

ansible hosts-group -m copy -a “src=/tmp/test1 dest=/tmp/test1” Here, copy module takes two parameter that is src and dest.

Explanation for Wrong answer:

* The option a, c, and d are not related with Ansible

1. **What is the purpose of the “Target” Section in Ansible Playbooks?**
2. **It specifies the target user**
3. **It specifies the target module**
4. **It specifies the target roles to run**
5. **It specifies the target inventory group**

The correct answer is option d - It specifies the target inventory group.

Explanation for correct answer:

* The Target section specifies the target host group defined in the Ansible Inventory File. It is used to select remote machines on which the Ansible tasks need to be applied.

Explanation for Wrong answer:

* The option a, b, and c are not related with Ansible Playbook Target section

**Module 4: Ansible - Till the End**

1. **\_\_\_\_\_\_\_\_\_\_\_\_are just like regular tasks in an Ansible playbook but they only run if the task contains a notify directive and also indicates that it changed something**
2. **Handlers**
3. **Task**
4. **Until**
5. **Notify**

The correct answer is option a - Handlers.

Explanation for correct answer:

Handlers are just like regular tasks in an Ansible playbook (see Tasks) but they only run if the Task contains a notify directive and also indicates that it changed something. For example, if a config file is changed, then the task referencing the config file templating operation may notify a service restart handler. This means services can be bounced only if they need to be restarted. Handlers can be used for things other than service restarts, but service restarts are the most common usage.

Explanation for Wrong answer:

* Option b - Playbooks exist to run tasks. Tasks combine an action (a module and its arguments) with a name and optionally some other keywords
* Option c - Until is used where we can prove a task or something that needs to be done until a certain condition is met, means it will work or run until that particular task is completed.
* Option d – Notify is the act of a task registering a change event and informing a handler task that another action needs to be run at the end of the play. If a handler is notified by multiple tasks, it will still be run only once. Handlers are run in the order they are listed, not in the order that they are notified.

1. **Is this statement true:**

**Ansible loops use a construct called with\_items which allows a particular task to be repeated for multiple items in a list.**

The correct answer is True.

Explanation for correct answer:

* Yes, this statement is true. Generally, Ansible is not a programming language. Loops prefers to be more declarative, though various constructs like with\_items allow a particular task to be repeated for multiple items in a list. Certain modules, like yum and apt, are actually optimized for this, and can install all packages given in those lists within a single transaction, dramatically speeding up total time to configuration.

Explanation for Wrong answer:

* No, this statement is true. Generally, Ansible is not a programming language. Loops prefers to be more declarative, though various constructs like with\_items allow a particular task to be repeated for multiple items in a list. Certain modules, like yum and apt, are actually optimized for this, and can install all packages given in those lists within a single transaction, dramatically speeding up total time to configuration.

1. **\_\_\_\_\_\_\_\_\_\_\_ is the preferred templating format for Ansible templates.**
2. **Jinja2**
3. **JSON**
4. **Inventory**
5. **Library**

The correct answer is option a – Jinja2.

Explanation for correct answer:

Jinja2 is the preferred templating format for Ansible templates. It is a very simple Python templating language which is generally readable and easy to write.

Explanation for Wrong answer:

* Option b - Ansible uses JSON for return data from remote modules. This allows modules to be written in any language, not just Python.
* Option c – Inventory is a A file (by default, Ansible uses a simple INI format) that describes Hosts and Groups in Ansible. Inventory can also be provided via an Inventory Script (sometimes called an “External Inventory Script”).
* Option d – Library is a collection of modules made available to /usr/bin/ansible or an Ansible playbook.

1. **Which keyword is used for conditional operations in tasks?**
2. **check**
3. **if**
4. **when**
5. **while**

The correct answer is option c - when.

Explanation for correct answer:

* When parameter is used to conditionally run a task - an optional conditional statement attached to a task that is used to determine if the task should run or not. If the expression following the when: keyword evaluates to false, the task will be ignored.

Example is

- name: Print conditionally

shell: uptime

when: ansible\_hostname == ‘node01’

This task will run only on node 01.

Explanation for Wrong answer:

* The option a, b and c are not related to Ansible Conditional

**Module 5: Deep Dive into Ansible Playbook**

1. **\_\_\_\_\_\_\_\_\_\_\_\_allows us to keep sensitive data such as passwords or keys in encrypted format rather than as plaintext.**
2. **Ansible Crypt**
3. **Ansible Arch**
4. **Ansible Vault**
5. **Ansible Cellar**

The correct answer is option c – Ansible Vault.

Explanation for correct answer:

* Ansible Vault allows us to keep sensitive data such as passwords or keys in encrypted files rather than as plaintext in the yaml file.

Explanation for Wrong answer:

* The option a, b and c are not related to Ansible

1. **Is this statement true:**

**We use “ansible-playbook” command to execute an Ansible playbook. But to run an Ansible ad-hoc command - we can use “ansible” followed by other input parameters.**

The correct answer is True.

Explanation for correct answer:

* Yes, this statement is true. To run a playbook, we use the Ansible command “ansible hyphen playbook” to invoke a playbook where as to run an Ad-hoc Command in this case, we can just type ansible followed by writing our commands.

Explanation for Wrong answer:

* No, this statement is true. To run a playbook, we used the Ansible command “ansible hyphen playbook” to invoke a playbook where as to run an Ad-hoc Command in this case, we can just type ansible followed by writing our commands.

1. **Which of the following statement is incorrect in the context of Ansible roles:**
2. **Ansible roles are units of organization in Ansible.**
3. **Assigning an Ansible role to a group of**[**hosts**](http://man.hubwiz.com/docset/Ansible.docset/Contents/Resources/Documents/docs.ansible.com/ansible/glossary.html#term-host)**(or a set of**[**groups**](http://man.hubwiz.com/docset/Ansible.docset/Contents/Resources/Documents/docs.ansible.com/ansible/glossary.html#term-group)**, or**[**host patterns**](http://man.hubwiz.com/docset/Ansible.docset/Contents/Resources/Documents/docs.ansible.com/ansible/glossary.html#term-globbing)**, etc.) implies that they should implement a specific behaviour.**
4. **An Ansible role may execute numerous Ansible tasks while applying certain variable values and**[**handlers**](http://man.hubwiz.com/docset/Ansible.docset/Contents/Resources/Documents/docs.ansible.com/ansible/glossary.html#term-handlers)
5. **None of the above**

The correct answer is option d - None of the above.

Explanation for correct answer:

* Yes, the option a, b and c are true. As Roles are units of organization in Ansible. Assigning a role to a group of hosts (or a set of groups, or host patterns, etc.) implies that they should implement a specific behaviour. A role may include applying certain variable values, certain tasks, and certain handlers – or just one or more of these things. Because of the file structure associated with a role, roles become redistributable units that allow you to share behaviour among playbooks – or even with other users.

Explanation for Wrong answer:

* No, the option a, b and c are true. As Roles are units of organization in Ansible. Assigning a role to a group of hosts (or a set of groups, or host patterns, etc.) implies that they should implement a specific behaviour. A role may include applying certain variable values, certain tasks, and certain handlers – or just one or more of these things. Because of the file structure associated with a role, roles become redistributable units that allow you to share behaviour among playbooks – or even with other users.

1. **\_\_\_\_\_\_\_\_\_\_\_\_ is a community site for finding, downloading, rating, and reviewing all kinds of community developed Ansible roles and can be a great way to get a jumpstart on your automation projects.**
2. **Ansible Galaxy**
3. **Ansible Assembly**
4. **Ansible Cluster**
5. **Ansible Collection**

The correct answer is option a – Ansible Galaxy.

Explanation for correct answer:

* Ansible Galaxy, is a free site for finding, downloading, rating, and reviewing all kinds of community developed Ansible roles and can be a great way to get a jumpstart on your automation projects.

Explanation for Wrong answer:

* The option b, c and d are not related to Ansible

1. **Which one of these files act as an entry point for Ansible roles’ individual folder?**
2. **start.yaml**
3. **main.yaml**
4. **startpoint.yaml**
5. **index.yaml**

The correct answer is option b - main.yml.

Explanation for correct answer:

* A task is place where when we first apply a role, it will come to and inside these folders, we should have a main.yml file which should be executed first and it will be in the location main.yml.

Explanation for Wrong answer:

* The option a, c and d are not related to Ansible Role

**Module 6: Managing Amazon Resources using Ansible**

1. **Is this statement true:**

**Ansible makes configuration management, application deployment and continuous delivery in AWS easy. But it needs agent software installed on the target hosts.**

The correct answer is False.

Explanation for correct answer:

* Yes, this statement is false. We can use Ansible to provision and manage our Amazon Web Services (AWS) Cloud infrastructure. Automated infrastructure provisioning and management is a key component of Continuous Delivery and Ansible makes configuration management, application deployment and continuous delivery in AWS easy, with no agents or special coding skills required. We can use Ansible for provision and manage our various AWS resources including EC2, VPC, RDS, S3, ELB, Auto Scaling, IAM and Route 53.

Explanation for Wrong answer:

* No, this statement is false. We can use Ansible to provision and manage our Amazon Web Services (AWS) Cloud infrastructure. Automated infrastructure provisioning and management is a key component of Continuous Delivery and Ansible makes configuration management, application deployment and continuous delivery in AWS easy, with no agents or special coding skills required. We can use Ansible for provision and manage our various AWS resources including EC2, VPC, RDS, S3, ELB, Auto Scaling, IAM and Route 53

1. **Is this statement true:**

**Ansible can deploy only one AWS EC2 instance at a time.**

The correct answer is False.

Explanation for correct answer:

* Yes, this statement is false. Using Ansible and AWS modules, we can create as many AWS EC2 instances as we want. While creating the EC2, in the command we have a count parameter which specify the number of instances to launch

Explanation for Wrong answer:

* No, this statement is false. we can create as many instance of EC2 servers as we want. While creating the EC2, in the command we have a count parameter which specify the number of instances to launch

1. **Which Ansible module can be used for provisioning of AWS EC2 instances**
2. **ec2**
3. **instance**
4. **ec2\_ansible**
5. **ansible\_instance**

The correct answer is option a – ec2.

Explanation for correct answer:

* ec2 module allows provisioning of EC2 instances. Provisioning will be against Ansible master server in a play that operates on localhost.

Explanation for Wrong answer:

* The option b, c and d are not related to Ansible.

1. **After the setup of your environment is completed. To test and ensure you can communicate with the AWS EC2 API, by executing which one of the following command:**
2. **aws ec2 describe-instances**
3. **aws ec2 decide-instances**
4. **aws ec2 subscribe-instances**
5. **aws ec2 define-instances**

The correct answer is option a - aws ec2 describe-instances.

Explanation for correct answer:

After the setup of your environment is completed, to test and ensure you can communicate with the AWS EC2 API, by executing the following command:

aws ec2 describe-instances

Explanation for Wrong answer:

* The option b, c and d are not related to Ansible.